Installation Guide

Schrödinger Suite 2006

Schrödinger Press

See the Copyright Notices for full copyright details.

To the maximum extent permitted by applicable law, this publication is provided "as is" without warranty of any kind. This publication may contain trademarks of other companies.

Please note that any third party programs ("Third Party Programs") or third party Web sites ("Linked Sites") referred to in this document may be subject to third party license agreements and fees. Schrödinger, LLC and its affiliates have no responsibility or liability, directly or indirectly, for the Third Party Programs or for the Linked Sites or for any damage or loss alleged to be caused by or in connection with use of or reliance thereon. Any warranties that we make regarding our own products and services do not apply to the Third Party Programs or Linked Sites, or to the interaction between, or interoperability of, our products and services and the Third Party Programs. Referrals and links to Third Party Programs and Linked Sites do not constitute an endorsement of such Third Party Programs or Linked Sites.

Revision A, April 2006

Contents

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, and screen output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

In descriptions of command syntax, the following UNIX conventions are used: braces `{ }` enclose a choice of required items, square brackets `[ ]` enclose optional items, and the bar symbol `|` separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Installation Notes

This document describes the installation of Maestro™ 7.5, CombiGlide™ 1.0, Epik™ 1.0, Glide™ 4.0, Impact™ 4.0, Jaguar™ 6.5, Liaison™ 4.0, LigPrep™ 2.0, MacroModel® 9.1, Phase™ 2.0, Prime™ 1.5, QikProp™ 2.5 (including QikFit™ and QikSim™), QSite™ 4.0, SiteMap™ 2.0, and Strike™ 1.5. Periodically, we release updates of our software. These minor releases are not automatically shipped on CD, but are posted on the Schrödinger Support Center. You are invited to download these updates for the version of the software package you have purchased. The instructions in this document refer to the release of April 2006. For earlier versions of this document, see the support page of our web site.

IMPORTANT!

- `gunzip` is required for installation of all Schrödinger software.
- The `libblas` library must be installed on SGI IRIX platforms.
- **Perl is required** to run Schrödinger software**. The minimum required version is 5.004.**
- Support for SGI mips3 executables has been discontinued. Support is provided for the SGI IRIX mips4 and SGI mips5 platforms, via the IRIX-mips4 installation.
- Support for AIX com executables has been discontinued. Support is provided for IBM AIX pwr3, pwr4 and pwr5 platforms, via the AIX-pwr3 installation.
- Support for the DQS queuing system has been discontinued.
- The environment variable `SCHRODINGER` must be set to the directory in which the software was installed before Schrödinger products can be launched.
- To run jobs, Schrödinger products must be installed on the submission host as well as the execution hosts.

If you have difficulty with the installation, please contact your system manager or Schrödinger (by phone at (503) 299-1150, or by e-mail at help@schrodinger.com).

Installation Process Summary

This is a summary of the installation process. For detailed instructions, see the page number provided in each step.

1. Check this guide for:
   - System requirements (page 50)
   - Disk space requirements (page 53)
   - Product-specific installation information (page 54-65)
2. Mount the CD (page 11) or download the software from the Schrödinger Support Center:

   If you download the software, extract the downloaded tar file:

   ```
   tar xvf Schrodinger_Internet_Download.tar
   ```
3. *Optional:* Prepare to install on remote machine (page 12).
4. Run the `platform` script to verify that your machine meets the system requirements (page 13).
5. Run the `INSTALL` script to install the products (page 14).
6. Set the `SCHRODINGER` and `DISPLAY` environment variables (page 21).
7. Obtain a license for the product(s):

   a. Obtain machine information (page 25).

   b. Request a license (page 25).

   c. Install the license (page 27).
8. *Optional:* Substitute run-time libraries (page 33).
9. *Optional:* Prepare for Job Submission (page 34).
10. *Optional:* Prepare for Batch Queue Submission (page 39).
11. Type `$SCHRODINGER/maestro` to launch Maestro (page 49).

Mounting the Product CD

These instructions explain how to mount a disc on each of the platforms for which Schrödinger software is supported.

Insert the product CD into the CD-ROM drive.

If the CD is not mounted automatically, mount it using the information below. If you are installing to a different machine from the one on which you mount the CD, particularly if you mount the CD on a Windows machine, you must also read the instructions on page 12. If you need further assistance, see your system administrator.

SGI IRIX

Usually CDs are mounted automatically on `/CDROM` (for the first CD-ROM drive). If the CD is not automatically mounted, consult your IRIX *Administration Guide* or your system administrator.

IBM AIX

If the CD is not automatically mounted, log in as `root` and enter the following command:

```
mount -o ro -v cdrfs /dev/device /mount-dir
```

where *device* is the CD-ROM drive (e.g. `cd0` for the first CD-ROM drive) and *mount-dir* is usually `cdrom`.

If you do not have root access, contact your system administrator.

Linux

In recent RedHat releases, CDs are automatically mounted on `/mnt/cdrom`. If the CD is not automatically mounted, log in as `root` and enter the following command:

```
mount /dev/cdrom
```

If you do not have root access, contact your system administrator.

Preparing to Install on a Remote Machine

Use these instructions only to install from a CD to a computer that does not have a CD-ROM drive.

- **If you are installing locally or to an NFS-mounted disk, skip this section and proceed to "Verifying System Compliance" on page 13.**
- **If you are installing from a download, you need only copy the downloaded tar file to the remote machine, extract it and proceed to "Verifying System Compliance" on page 13.**

To prepare for installation of Schrödinger products on a remote machine:

1. Mount the product CD on the local machine, as described on page 11.
2. Change to the mount directory and display the CD contents.
3. Copy the following files to the remote machine's hard drive:
   - product tar files for your platform
   - `INSTALL` file
   - scripts in the top-level directory
   - Maestro, Python, and mmshare tar files for your platform
   - data tar files for your product
   - third-party software and databases for Prime
   - *optional:* documentation tar files
4. Change file names if necessary.

   If you are installing from a CD-ROM drive on a Windows machine, Windows may change the case of the file names. The tar files and `platform` script should be in lower case and the `INSTALL` script and the `README` file should be in upper case. Use the `TRANS.TBL` file to rename the files with the correct case.
5. Log in on the remote machine, change to the directory containing the copied files, and proceed to "Verifying System Compliance" on page 13.

Note: For remote jobs, Schrödinger products must be installed on the submission host as well as the execution hosts.

Verifying System Compliance

Before continuing with the installation, verify that your system satisfies the minimum requirements to run Schrödinger software.

1. Locate the `platform` script:
   - top-level directory on the CD
   - *download-directory*`/Schrodinger_Internet_Download`
   - directory on the remote machine containing the copied Schrödinger files
2. Copy the script to your local hard drive or to each host that will run the Schrödinger software:
   - *Shared drive*—copy the script to a shared disk.
   - *Remote machine*—copy the script to the remote machine.
3. Log on to each host on which you plan to run Schrödinger software and change to the directory that contains the script.
4. Enter the following command:

   ```
   ./platform -s
   ```

   The script indicates whether your system meets the requirements or needs to be updated. If you receive an error message, postpone installation of your Schrödinger software until you have updated your system. For help obtaining any missing libraries, see the appropriate product-specific section of this guide.
5. *Optional:* If you plan to install executables intended for platforms or machine types other than that on which the CD is mounted, run the platform script without options:

   ```
   ./platform
   ```

   and make note of the recommended version, so you can choose the correct version during the installation.
6. *Optional:* To see a summary of the platform information, enter:

   ```
   ./platform -l
   ```

   The script checks the operating system and distribution, CPU type, number of processors, perl version number, and relevant libraries (`xlf` for IBM-AIX, `libblas` for SGI and `glibc` for Linux).

Installing the Products

Before installing Schrödinger products and documentation, read "Hardware and Software Requirements" on page 50 and the product-specific installation instructions on pages 54-65. If you are installing on a cluster, read "Configuring Clusters" on page 44.

When you select the locations for installing the software, you must ensure that the software is accessible from all hosts that are used either to submit jobs or to run jobs. This includes individual nodes on a cluster.

Note: The installation process does not replace the `$SCHRODINGER/schrodinger.hosts` or `$SCHRODINGER/license` files or files in the `$SCHRODINGER/queues` directory. If you want to install new versions of these files, you must move or remove them first.

1. Change to the directory that contains the Schrödinger software:
   - CD-ROM mount directory
   - directory on the remote machine containing the copied files
   - *download-directory*`/Schrodinger_Internet_Download`

     If you downloaded the files, extract the tar file in the download directory first:

     ```
     tar xvf Schrodinger_Internet_Download.tar
     ```
2. Enter the command

   ```
   ./INSTALL
   ```
3. Enter the information requested by the `INSTALL` script.
   - You can accept the default values for each question by pressing RETURN
   - You can quit the `INSTALL` script at any time by pressing CTRL-C.
   - If you realize you have entered incorrect information, simply press RETURN at all of the prompts, then type `n` at the confirmation screen to start the questions again.

   Below are explanations of the questions asked by the script:

   a. **`SCHRODINGER` directory:** This is the installation directory, where the executables, data files, and other files related to Schrödinger products will be installed. Depending on the type of license you have (see page 24), we recommend the following installations:

- *Token-based or IP-based license:* Use a shared file system so that you only have to install the software once and all client machines with access can use it.
- *Node-locked license:* Use the local file system of the machine that will run the software or an NFS-mounted file system (for example, if your local file system does not have enough free space to install the software).

The INSTALL script can use an existing SCHRODINGER directory or create a new one. When the script has located or created the SCHRODINGER directory, it asks you to confirm that the selection is correct. Press RETURN to accept.

b. **Hardware/Software platform:** In this screen, the INSTALL script recommends the most compatible version of the executables for the current machine, based on the machine type and operating system. Press RETURN to continue. (You will select the products in the next screen.)

If you plan to install Schrödinger software on a machine other than that on which the INSTALL script is running, copy the platform script to that machine, log in to it and run the script without options:

```
./platform
```

Make a note of the recommended version so you can select it on the next screen of the installation.

c. **Product selection:** This screen lists all the modules available for installation. Those that are compatible with the current machine are marked with a yes in the compatible column.

1. To determine which modules you need, see Table 1. For disk space requirements, see Table 4 on page 53.

   The lists in Table 1 include only the modules required for the particular product or solution. If you want to use a product that is included in part of another product or solution, you must select the modules that are listed for the product. For example, if you install Induced Fit Docking, and want to run Prime separately, you must include the modules on the Prime list as well as those on the Induced Fit Docking list. If you want to set up jobs from Maestro, you must also install Maestro; for some products you must set up jobs from Maestro.

2. To select product or documentation modules, enter the index numbers (e.g. 1,2,3-5) and press RETURN to redisplay the list with INSTALL in the action column for the products you selected.
3. You can then select more products or type none to start over.
4. When you have finished, press RETURN to accept the current selections.

If you are installing documentation, you should install the Maestro and general documentation as well as that listed for the product.

Note: For Prime, the data module is split across two CDs. If you install the second CD first, the first part of the data module will not be installed. You can install it later with the following command:

```
./INSTALL psp-vversion-data-1of2.tar.gz
```

This command starts the install script with the data module selected, and skips the selection stage. You must also use this command to install the data module if you install Prime after installing PrimeCM.

Table 1. Product and platform selections for installation of Schrödinger products. All required modules are included on the product CDs or in the download.

Schrödinger Product	Modules to Install		
CombiGlide	`combiglide`	*version*	*platform*
	`combiglide`	*version*	`<docs>`
	`glide`	*version*	`<docs>`
	`impact`	*version*	*platform*
	`ligrep`	*version*	`<docs>`
	`mmacromodel`	*version*	*platform*
	`qikprop`	*version*	*platform*
	`services`	*version*	*platform*
Epik	`epik`	*version*	`<docs>`
	`epik`	*version*	*platform*
Glide	`glide`	*version*	`<docs>`
	`impact`	*version*	*platform*
	`impact`	*version*	`<docs>`
	`services`	*version*	*platform*
Induced Fit Docking[a]	`glide`	*version*	`<docs>`
	`impact`	*version*	*platform*
	`inducedfit`	*version*	`<docs>`
	`primeCM`	*version*	*platform*
	`prime`	*version*	`<docs>`
	`services`	*version*	*platform*
Jaguar	`jaguar`	*version*	*platform*
	`jaguar`	*version*	`<docs>`
Liaison	`liaison`	*version*	`<docs>`
	`impact`	*version*	*platform*
	`impact`	*version*	`<docs>`
	`strike`	*version*	`<docs>`

Table 1. Product and platform selections for installation of Schrödinger products. All required modules are included on the product CDs or in the download.

Schrödinger Product	Modules to Install		
Ligand & Structure-Based Descriptors[b]	liaison	*version*	<docs>
	impact	*version*	*platform*
	impact	*version*	<docs>
	lsbd	*version*	<docs>
	macromodel	*version*	*platform*
	macromodel	*version*	<docs>
	primeCM	*version*	*platform*
	prime	*version*	<docs>
	qikprop[c]	*version*	*platform*
	qikprop	*version*	<docs>
LigPrep	ligrep	*version*	<docs>
	macromodel	*version*	*platform*
	macromodel	*version*	<docs>
	services	*version*	*platform*
MacroModel	macromodel	*version*	*platform*
	macromodel	*version*	<docs>
Maestro	maestro	*version*	*platform*
	maestro	*version*	<docs>
Phase	phase	*version*	*platform*
	phase	*version*	<docs>
	ligprep	*version*	<docs>
	macromodel	*version*	*platform*
	macromodel	*version*	<docs>
	services	*version*	*platform*
Prime[d]	psp	*version*	*platform*
	psp	*version*	<data>[e]
	prime	*version*	<docs>
	pdb		<database>
	blast		*platform*
	blast		<database>
	hmmerpfam		*platform*
	hmmerpfam		<database>
Prime-CM[f]	primeCM	*version*	*platform*
	prime	*version*	<docs>
	blast		*platform*
	hmmerpfam		*platform*
QikProp[g]	qikprop	*version*	*platform*
	qikprop	*version*	<docs>

Table 1. Product and platform selections for installation of Schrödinger products. All required modules are included on the product CDs or in the download.

Schrödinger Product	Modules to Install		
QM-Polarized Ligand Docking	`impact`	*version*	*platform*
	`glide`	*version*	`<docs>`
	`jaguar`	*version*	*platform*
	`qpld`	*version*	`<docs>`
	`qsite`	*version*	`<docs>`
QSite	`impact`	*version*	*platform*
	`impact`	*version*	`<docs>`
	`jaguar`	*version*	*platform*
	`jaguar`	*version*	`<docs>`
	`qsite`	*version*	`<docs>`
SiteMap	`sitemap`	*version*	`<docs>`
	`impact`	*version*	*platform*
Strike	`strike`	*version*	`<docs>`
	`maestro`	*version*	*platform*
	`maestro`	*version*	`<docs>`
Virtual Screening Workflow	`glide`	*version*	`<docs>`
	`impact`	*version*	*platform*
	`ligprep`	*version*	`<docs>`
	`macromodel`	*version*	*platform*
	`macromodel`	*version*	`<docs>`
	`qikprop`	*version*	*platform*
	`qikprop`	*version*	`<docs>`
	`services`	*version*	*platform*
	`vsw`	*version*	`<docs>`

a. Induced Fit Docking does not require installation of third party software or databases.

b. This list includes all products that can be used to generate descriptors. If you only want descriptors for some products, install only those products.

c. Optional module. If not installed, filtering based on properties is not available.

d. The modules are distributed over several CDs. You should install the CDs in order, otherwise the data installation will not be complete.

e. This module only appears if you are installing from the second CD. You must select this module to ensure that the data installation is complete.

f. You must also install the third party databases—see "Required Third-Party Software and Databases" on page 63. These databases are not provides with the download.

g. Includes QikFit and QikSim.

d. **Scratch directory:** This directory is for the large, temporary files generated by computational programs during calculations. We recommend this directory be located on a fast, local drive with at least 4 GB of disk space. The `INSTALL` script checks for existing directories named `/scr`, `/scratch` or `/usr/tmp` and suggests the first of these as the default.

   If you decide to use a different directory, you will need to create it first. The `INSTALL` script will not create it for you. Also, make sure each person who wants to run jobs has write access to the scratch directory.

   Once you have specified a scratch directory, the `INSTALL` script adds a `localhost` entry to the `schrodinger.hosts` file as follows:

   - If a `schrodinger.hosts` file already exists and contains a `localhost` entry, no action is taken, even if there is no `tmpdir` setting in the `localhost` entry. You will need to add the `tmpdir` setting manually (see "Modifying the Hosts File" on page 34).
   - If a `schrodinger.hosts` file already exists but it does not contain a `localhost` entry, a `localhost` entry is added with a `tmpdir` setting.
   - If a `schrodinger.hosts` file does not exist, the script creates the file with just a `localhost` entry and `tmpdir` setting.

4. Confirm the information you provided.

   When you have finished entering the information, the `INSTALL` script summarizes your choices. In addition to the products you specified, the product `mmshare` is listed and installed since it is needed to run all Schrödinger software. If any of the summary information is incorrect, answer "`n`" at the prompt to run through the questions again. Once you are satisfied with your answers, press RETURN to install the software. The installation can take several minutes. Prime installation, including third-party software and databases, can take 20 minutes.

5. Record the `machid` information and copy it into an e-mail.

   When the installation is complete, the `INSTALL` script runs the `machid` program, which generates machine-specific information about the computer on which it is run. Copy this information into an e-mail to request a license for your Schrödinger software. See "Obtaining a License" on page 24 for full details on how to request a license.

   If the executables you installed are intended for platforms other than that on which the CD is mounted, `machid` fails. You will need to log in to the machine on which you plan to run the Schrödinger software run `machid` from there. See "Obtaining Machine Information" on page 25.

6. Remove temporary installation directories and files. If you copied tar files onto a remote machine, delete those files now.

Repeat this procedure for all machines on which you want to use the software. Once you have installed the software, you must obtain a license to run it. See “Obtaining a License” on page 24.

Setting the Environment Variables

Before you can launch Schrödinger software, you must set some environment variables. In addition to those listed below, there may be product-specific environment variables that need to be set—see the requirements section for each product.

`SCHRODINGER`	required for all Schrödinger products
`DISPLAY`	required for Maestro (may be automatically set on login)

To set the `SCHRODINGER` environment variable, enter the following command, replacing *install-directory* with the full installation path (for example, `/software/Schrodinger`):

csh, tcsh:	`setenv SCHRODINGER` *install-directory*
bash, ksh:	`export SCHRODINGER=`*install-directory*

You can add this command to your shell startup script file (`.cshrc` or `.login` for csh, `.profile` for sh or ksh, `.bashrc` for bash). Otherwise, you must set the `SCHRODINGER` environment variable each time you launch Schrödinger software from a new shell.

We recommend that you add `$SCHRODINGER` and `$SCHRODINGER/utilities` to your `PATH` definitions. Many utility programs have been centralized into the `$SCHRODINGER/utilities` directory. Among other things, these utilities perform structure format conversion, protein preparation, Glide pose re-ranking, and parallel Glide job submission from the command line.

To set the `DISPLAY` environment variable, enter the following command, replacing *machine-name* with the name of the display machine.

csh, tcsh:	`setenv DISPLAY` *machine-name*`:0.0`
bash, ksh:	`export DISPLAY=`*machine-name*`:0.0`

To determine the name of a display machine, enter the command `hostname`.

If you expect either long delays when a program tries to obtain a license token, or competition between programs for license tokens, you can set the time limit for trying to obtain a license token in the `SCHRODINGER_LICENSE_RETRY` environment variable. This environment variable can be set to time values, such as 300s, 10m, 2h, or to an integer value, which is interpreted as a time in seconds. The default is 10 minutes.

If you plan to run jobs remotely and use `ssh`, you will also need to set the `SCHRODINGER_RSH` environment variable—see “Setting up Passwordless ssh Access to Remote Hosts” on page 37.

You can also set environment variables for each host in the `schrodinger.hosts` file. See “Modifying the Hosts File” on page 34 for more information.

Documentation

Documentation is supplied in PDF format in gzipped tar files. There is a tar file for each software product and a `general` tar file, which contains information applicable to multiple products, including this document.

If you did not install the documentation when you installed the product, you can do so by running the `INSTALL` script again and selecting the desired documentation. The documentation is installed in the directory `$SCHRODINGER/docs/`*product*, where *product* is the product name, such as `maestro`, and `$SCHRODINGER` is the installation directory.

The documentation includes a master index, providing links to all manuals for all Schrödinger products. This table of contents is available in both PDF and HTML format:

```
$SCHRODINGER/docs/index.html
$SCHRODINGER/docs/schrodinger_documentation.pdf
```

In older browsers, the links in the PDF file can fail if you open it via a plug-in. If this happens, you should open the file directly in your PDF reader. The PDF index can be opened from Maestro by choosing Manuals from the Help menu.

The documentation also includes a search index, named `schrodinger.pdx`, which is included with the general documentation and is installed into `$SCHRODINGER/docs`. You can use this search index in Acrobat Reader 6.0 and 7.0 to search for text across the entire documentation set. Instructions are given in `schrodinger_documentation.pdf`.

Documentation is also available from the Support Center of our web site. In addition to the manuals in PDF format, the web site contains FAQ pages for general information and for each product. Information on known issues can be found on the FAQ pages.

Note: The latest version of the documentation is posted on the web site. If you suspect that there is an error in the documentation, check for corrections and additions on the documentation page.

Obtaining a License

To obtain a license:

1. Determine your license type (page 24).
2. Run `machid` to collect your machine information (page 25).
3. E-mail the machine information to Schrödinger to request the license (page 25).
4. Install the license codes (page 27).

Schrödinger UNIX products use FLEXlm licenses. If you have questions about the FLEXlm license manager daemon `lmgrd`, consult the latest version of the FLEXlm Users' Guide at:

http://www.macrovision.com/services/support/flexlm/enduser.pdf [1]

Determining Your License Type

Schrödinger issues the following types of licenses:

Token-based (product-specific)	Allows jobs for a specific product to be run on any machine, but only up to the total number of jobs specified in the license code.
Token-based (interchangeable):	Allows jobs for any product listed in the license code to be run on any machine, but only up to the total number of jobs specified in the license code.
IP-based (with server restriction)	Allows the software to be run on any machine whose IP address falls in the private ranges 192.168.*.*, 10.*.*.*, and 172.16.*.* through 172.31.*.*.
IP-based (subnet)	Allows the software to be run on any machine whose IP address falls in the address range specified in the license code.
Node-locked	Allows the software to be run on a single, specific machine.

Note: Schrödinger uses the term "token", while FLEXlm uses the word "license". In the context of obtaining your license, both words mean the same thing.

[1] Please see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

The following licenses require a license server:

- Token-based (both product-specific and interchangeable)
- IP-based (with server restriction)

The license server does not need to be a particularly powerful machine, as the license server daemon is a lightweight process. However, the license server does need to be accessible over the network to any machines that can check out licenses, so you should choose a machine that has good network connectivity and is not frequently shut down or rebooted.

Obtaining Machine Information

To obtain machine information, run the `machid` program, located in the installation directory:

Token-based license: Run the `machid` command on the machine designated as the license server. If you wish to run in redundant-server mode, send us the `machid` output from 3 machines and specify which should be the primary server.

IP-based license: Run the `machid` command on one representative machine in each of the IP-subnets in which you plan to run the software. It is not necessary to send us the `machid` output for every machine in each subnet. If you are using certain private IP-subnets (see previous page for a list), you must also send the `machid` output for the machine you have chosen as a license server.

Node-locked license: Run the `machid` command on the machine where the software will be run. Please check very carefully that the `machid` command is executed on the machine where you plan to run the software, as we use this information to generate a single license for that machine only.

Multiple licenses: Run the `machid` command on each machine on which you installed the software, copy the output from each machine and send that output to Schrödinger as described below.

Requesting a License

To request a license from Schrödinger:

1. Copy the output from the `machid` program, between the `start of machid output` and `end of machid output` lines, and paste it into an e-mail message.
2. Append the name of your organization, the Schrödinger software purchaser, and the primary software user.
3. Send this information to help@schrodinger.com.

Once Schrödinger receives your request, we will generate your license code and send it to you via e-mail, usually within one business day.

Explanation of License Codes

Your license is sent to you by e-mail in the form of an attachment. Below is a full example of a token-based license code. The other keys have slight differences and are listed in the following sections.

Token-Based License Code (Product-Specific)

```
SERVER firth 690571cd
VENDOR SCHROD
INCREMENT IMPACT_MAIN SCHROD 30 31-May-2005 42 HOSTID=ANY SUPERSEDE \
        ISSUED=14-Jun-2004 SIGN="0444 4239 EBF0 A6D2 686F 0E21 5F30 \
        3067 186E 6F45 5E82 9193 66F8 2130 BFFC 1701 52E7 2926 4F5D \
        40FF 8C2F 6DBA DD9F 07E4 3259 A17E 6ADC C2AB 0778 5676"
```

This example contains the following elements:

Server:	`SERVER firth 690571cd`
Vendor:	`VENDOR SCHROD`
Increment or Feature:	(start of new license code)
Module:	`IMPACT_MAIN`
Vendor:	`SCHROD`
Version:	`30` (Impact 3.0)
Expiration Date:	`31-May-2005`
Number of Tokens:	`42`
Host ID:	`HOSTID=ANY`
Issue Date:	`ISSUED=14-Jun-2004`
License Code:	`SIGN="0444 ..."`

Token-Based License Code (Interchangeable)

Included modules:	`PACKAGE SUITE SCHROD COMPONENTS="PSP_SSP:14 ..."`
Options:	`OPTIONS=SUITE`
Number of shared tokens:	`INCREMENT SUITE SCHROD 10`

IP-Based License Code (with Server Restriction)

Server:	`SERVER melix 000d613b40dc`
Host ID range:	`HOSTID=INTERNET=192.168.0.*`

IP-Based License Code (Subnet)

Server:	no server listed
Number of tokens:	`uncounted`
Host ID range:	`HOSTID=INTERNET=*.*.*.*`

Node-Locked License Code

Server:	no server listed
Number of tokens:	`uncounted`
Host ID (one machine):	`HOSTID=000ea681ad36`

Installing the License

Schrödinger licenses are stored in the license file (`$SCHRODINGER/license`). This file may contain multiple license codes (e.g. for demos, multiple machines, etc.). Schrödinger programs identify and use the appropriate current license code. However, we recommend removing expired license codes from the license file. If you want to keep them for archival purposes, make sure that the active license codes are earlier in the file.

To install the license:

1. Copy the license codes from the e-mail attachment and paste it into your license file (`$SCHRODINGER/license`). Make sure there is an end quote and a carriage return at the end of each license code.
2. Save the changes to the license file and close it.
3. Check that the license file has the appropriate read permissions.

   If you have a node-locked license, your installation is complete.
4. Copy the license file to the appropriate hosts:
   - Token-based license: place a copy of (or symbolic link to) the license file in the `$SCHRODINGER` directory of each host listed on a `SERVER` line.
   - IP-based license with server restriction (there should be a `SERVER` line in the license code): place a copy of (or symbolic link to) the license file in the `$SCHRODINGER` directory of each host listed on a `SERVER` line.
   - IP-based license for subnet (there should be no `SERVER` line in the license code): place a copy of (or symbolic link to) the license file in the `$SCHRODINGER` directory of any other hosts that fall in the IP address range specified by the `HOSTID=INTERNET=` lines.

5. Start the license server daemon (page 28) or, if the license server daemon is already running, update it (page 29).

   If you have a FLEXlm license server running for other software, you will need to integrate the Schrödinger licensing with the existing licensing process. This includes defining LM_LICENSE_FILE.

6. Set up the client machines (page 29).

7. Enable license communication across a firewall or proxy (page 29).

If you encounter any problems, see "Troubleshooting" on page 30.

Starting the License Server Daemon

To start the license server daemon, lmgrd, enter the command:

```
$SCHRODINGER/licadmin SERVERUP -l lmgrd.`hostname`.log
```

lmgrd listens for license requests on the port designated by the third argument after the word SERVER on the SERVER line of the license file. For example, in the following SERVER line:

```
SERVER lsnode b0019732 27000
```

27000 is the port on which the machine lsnode listens for license requests. If no port is specified on the SERVER line, a default port in the range 27000-27009 is used.

If you have requested a license that allows you to run on three redundant servers, you must execute the command:

```
$SCHRODINGER/licadmin SERVERUP -l lmgrd.`hostname`.log
```

on each of the three servers. A port must be specified on each of the three SERVER lines. In most cases a port is already included in the license code. This port may be changed if the default ports specified in the license code are already in use on the machines acting as the redundant servers.

To see usage information for the licadmin utility, enter the command:

```
$SCHRODINGER/licadmin HELP
```

Updating the License Server Daemon

If the license server daemon is already running, enter the following command to instruct `lmgrd` to reread the license file:

```
$SCHRODINGER/licadmin REREAD
```

You must execute this command each time you make a change to the license file.

To check the status of available licenses, enter the following command:

```
$SCHRODINGER/licadmin STAT
```

Setting Up Client Machines

If the client machines do not have access to the license file directly (e.g. if the license file is on a local disk) the `LM_LICENSE_FILE` environment variable must be set on the client machines as follows:

csh, tcsh:	`setenv LM_LICENSE_FILE [`*port*`]@`*host*
bash, ksh:	`export LM_LICENSE_FILE=[`*port*`]@`*host*

In the commands above, *host* is the host name of the machine on which `lmgrd` is running, and *port* is the port number that is specified as the third argument after the word `SERVER` on the `SERVER` line of the license file. If no port is specified on the `SERVER` line, or if the port is in the default range of 27000-27009, then the value for *port* may be omitted. If `LM_LICENSE_FILE` is already defined and does not include this host (for example if you have other software that uses FLEXlm licensing), you can add to the definition as follows:

csh, tcsh:	`setenv LM_LICENSE_FILE $LM_LICENSE_FILE:[`*port*`]@`*host*
bash, ksh:	`export LM_LICENSE_FILE=$LM_LICENSE_FILE:[`*port*`]@`*host*

Enabling License Communication Across a Firewall or Proxy

If the client machine and the license server are separated by a firewall or proxy, you may need to specify a port on the `DAEMON` line of the license code in order to enable communication. Any unused port may be chosen, but the port specified must be made available on the firewall or proxy. For example, if the license code looks like the following:

```
SERVER lsnode b0019732 27000
DAEMON SCHROD PORT=10081
```

and the client machine and license server are separated by a firewall or proxy, then ports 27000 and 10081 must be made available to the client.

Troubleshooting

If you experience problems installing the license, check that the read permissions are set appropriately on the license file.

The formatting of the license file is important. The following command can be used to check for formatting or content errors:

```
$SCHRODINGER/licadmin CKSUM
```

Token-based licenses

Token-based licenses limit the number of instances of specific features of the program that may be used simultaneously. The `licadmin` utility can be used to check how many tokens are available for a specific `FEATURE` or `INCREMENT`. From the license server, enter the command:

```
$SCHRODINGER/licadmin STAT
```

If you are logged in to a client machine, enter the command:

```
$SCHRODINGER/licadmin STAT -c $LM_LICENSE_FILE
```

IP-based licenses

IP-based licenses are restricted by IP address and have the words `HOSTID=INTERNET=` in the `FEATURE` or `INCREMENT` section of the license code. If you have difficulty obtaining a license from a client machine, ensure that the client machine falls within the IP address subnet specified for that feature in the license code.

Node-locked licenses

Node-locked licenses are restricted to one specific machine, identified by `HOSTID=`*alphanum*. Node-locked features can only be used on the machine whose `HOSTID` value matches the `HOSTID` value specified for that feature in the license code.

Requesting assistance

If you have difficulties installing or using the license on the license server machine:

1. Set the `FLEXLM_DIAGNOSTICS` environment variable as follows:

csh, tcsh:	`setenv FLEXLM_DIAGNOSTICS 3`
bash, ksh:	`export FLEXLM_DIAGNOSTICS=3`

2. Run the following commands on the machine where the license file is installed (or where `lmgrd` is running, if you are using a license server).

```
echo $SHELL
hostname
whoami
pwd
echo $SCHRODINGER
$SCHRODINGER/machid
ls -l $SCHRODINGER/license
cat -v $SCHRODINGER/license
cat $SCHRODINGER/lmgrd.log
$SCHRODINGER/licadmin CKSUM
$SCHRODINGER/licadmin STAT
$SCHRODINGER/licadmin DIAG
ps -ef|egrep 'SCHROD|lmgrd'
cat $SCHRODINGER/schrodinger.hosts
$SCHRODINGER/hunt -rtest
cat /etc/hosts
which perl
perl -V
env |grep -i perl
```

3. Send the output and the `.log` file (if any) generated by the failed job to help@schrodinger.com:

If you have difficulties obtaining a license from the server on client machine, do the following:

1. Test whether the client is able to connect to the license server using other protocols (such as `telnet`, `ssh`, or `ping`).

2. Check with your system administrator to determine if a firewall is present between the client and the license server. If so, follow the instructions in the section "Enabling License Communication Across a Firewall or Proxy" on page 29.

3. Run the following commands on the client machine, and send the output to help@schrodinger.com:

```
echo $SHELL
hostname
whoami
pwd
echo $SCHRODINGER
$SCHRODINGER/machid
echo $LM_LICENSE_FILE
$SCHRODINGER/licadmin STAT -c $LM_LICENSE_FILE
```

```
$SCHRODINGER/licadmin DIAG -n -c $LM_LICENSE_FILE
cat $SCHRODINGER/schrodinger.hosts
$SCHRODINGER/hunt -rtest
nslookup ‘hostname‘
nslookup license-server-name
cat /etc/resolv.conf
cat /etc/host.conf
echo $RESOLV_SERV_ORDER
/sbin/ifconfig
cat /etc/hosts
which perl
perl -V
env |grep -i perl
```

Runtime Libraries

Schrödinger products are distributed as dynamically linked executables, with certain requisite dynamic libraries provided in the distributions. There are several reasons for this:

- Dynamic linking allows easy user-implementation of hardware-accelerated OpenGL on Linux. See below for more information.
- Dynamic linking simplifies the process of updating a particular library.
- Some of the libraries used in Schrödinger products are covered by the LGPL license, which stipulates, among other things, that our software be distributed in such a manner that end-user library modifications can be linked with our code. Distributing shared libraries, which are loaded at run-time, allows you to "plug in" your own compiled library replacements.

The libraries used by Schrödinger software products are stored in the directories:

`$SCHRODINGER/`*product*`/lib/`*platform*

where *product* is the product name and version number, and *platform* describes the platform and operating system. When a Schrödinger software program is launched, the startup script sets the appropriate environment variable so that the dynamic linker can locate the necessary libraries. This ensures that the library versions provided in the distribution are used in lieu of equivalents resident in the system.

To use a system library instead of the Schrödinger library, move the Schrödinger library from:

`$SCHRODINGER/`*product*`/lib/`*platform*

to:

`$SCHRODINGER/`*product*`/disabled_lib/`*platform*

The exception to the library search path is the graphics libraries. The library versions provided by the system are searched first, then the Schrödinger libraries, so that any library that is installed to take advantage of hardware graphics capabilities is used by default. If Maestro fails to find the OpenGL library in the shared library search path, the library in `$SCHRODINGER/maestro-v`*version*`/lib/linux-x86/gl` is used. To force the use of this library, launch Maestro with the `-SGL` option, or set the environment variable `SCHRODINGER_GL` to a non-null value.

Preparing for Job Submission

Schrödinger products use a common Job Control facility, which allows the user to submit, monitor, suspend and terminate jobs. To run jobs on the local host only, no additional configuration is needed. However, to permit users to run distributed jobs, run jobs on remote hosts, or submit jobs to batch queues you need to modify the hosts file (`schrodinger.hosts`). For remote job submission you also need to enable access to remote hosts using the `rsh` command or the `ssh` command without specifying a password (see page 36 and page 37). For batch queues, additional configuration is needed (see page 39). The procedures below and the Job Control facility itself are described in detail in the *Job Control Guide*.

Users who wish to configure job submission for themselves should make a copy of `schrodinger.hosts` in the directory `$HOME/.schrodinger` and edit it. User configuration is necessary if the user has a different user name on any host on which Schrödinger products will be run.

Modifying the Hosts File

To add a remote host for job submission:

1. Open the `$SCHRODINGER/schrodinger.hosts` file.
2. Create or modify an entry for each remote host using the keywords in Table 2 on page 35.
   - Syntax for the settings is *keyword*`:` *value.*
   - Keywords are case-insensitive.
   - Individual nodes in a cluster must be included unless they are only used as part of a properly-configured batch system.
   - Each entry must begin with a `name` setting.
   - Comments can be included by beginning a line with a # sign.
   - Entries for batch queue submission (including queues on clusters) must specify a temporary directory that is available on all compute nodes, and is writable by all valid users. This directory can be on a common filesystem shared by the nodes, or it can be identically-named local storage space on each node.
3. Save and close the file.

Table 2. Keywords for `schrodinger.hosts` *file settings.*

Keyword	Description
`name`	The name of the host entry or of the batch queue. For a host this is usually the host name. This name is displayed in Maestro by job control. The value `localhost` is a special value that means the host on which the job is launched.
`host`	The host name. This entry is only needed if it is different from `name` or if the batch queueing software is only available on a particular host.
`schrodinger`	The path to the Schrödinger software installation on the host.
`user`	The user name to use on the host. This should never be set in the hosts file in the installation directory. It is required if the user has a different user name on the defined host than on the host on which the job is launched.
`processors`	The number of processors available on the host. If the host is part of a cluster, this number should be the total number of processors available on the cluster. The default is 1.
`env`	Environment variables to be set on the host. The syntax for the environment variables is *variable*=*value*, regardless of the shell used. List each environment variable on a separate `env` line.
`tmpdir`	Base directory for temporary or scratch files. The file system on which this directory is mounted should be large enough for the largest temporary files, should be mounted locally, and should be writable by the user. Do not use symbolic links, as these can cause some programs to fail. The actual directory created for scratch files is */tmpdir/username/jobname*, where *tmpdir* is the directory defined here and *username* is the user name. Multiple `tmpdir` settings can be added for a given host and are used by Maestro, but the first setting is used otherwise.
`queue`	Queuing system name. PBS, SGE and LSF are the three supported systems.
`qargs`	Arguments to be used when submitting jobs to a batch queue. These arguments should specify any parameters that define the queue.

4. *Optional:* Test the `schrodinger.hosts` file by using the command:

   ```
   $SCHRODINGER/hunt -rtest
   ```

   This command tests the first `schrodinger.hosts` file found in the list below, and attempts to contact each of the hosts listed. Running this command serves as a check on host accessibility and on whether remote login has been set up appropriately on the hosts.

   - The file specified by the environment variable `SCHRODINGER_HOSTS`.
   - The `schrodinger.hosts` file in the current directory.
   - The `schrodinger hosts` file in `$HOME/.schrodinger`.
   - The `schrodinger.hosts` file in `$SCHRODINGER`.

Note: Default values for all hosts are taken from the entry for `localhost`.

A sample `schrodinger.hosts` file is shown below.

```
# Schrodinger hosts file
#
name:          localhost
schrodinger:   /software/schrodinger
tmpdir:        /scr
#
name:         larry
name:         curly
name:         moe
#
name:         server
schrodinger: /usr/local/schrodinger
tmpdir:       /big_scr
processors:  8
#
name:         cluster
host:         manager
queue:        PBS
qargs:        -lwalltime=1000:00:00
schrodinger: /sw/schrodinger
env:          SCHRODINGER_THIRDPARTY=/fast/disk
processors:  16
tmpdir:       /storage/TMPDIR
#
# End of Schrodinger hosts file
```

Setting Up Passwordless rsh Access to Remote Hosts

Access to remote hosts can be set up with `rsh` as described below. However, the use of `ssh` is preferred, both for security reasons and for its greater flexibility.

To set up passwordless `rsh` access, do one of the following:

- Create or modify the `hosts.equiv` file in the `/etc` directory on each host. This file should contain a list of hosts from which users can log in without giving their passwords (provided that their user names are the same on each of the machines). Creating a `hosts.equiv` file usually requires root permission.
- Create or modify the `.rhosts` file in the user's home directory on each of the remote hosts. The `.rhosts` file should list the names of the hosts and the user name used to log

in without specifying a password. The list should contain two lines for each machine—one with the machine name alone and one with the fully qualified name, as follows:

machine username-on-machine
machine.*domain username-on-machine*

The *username* in the `.rhosts` file is optional if the user name is the same on the remote hosts.

Note: Make sure to include an entry for each node on each cluster on which you plan to run jobs.

You do not need root permission to configure this file, but you must make sure that the file does not have "group" or "other" write permission. To ensure the correct permissions, use one of the following commands

```
chmod 644 $HOME/.rhosts
chmod go-w $HOME/.rhosts
```

You might also want to remove read permissions for "group" and "other".

Once you have configured `hosts.equiv` or `.rhosts`, use the following command to check for successful communication between the host that the job will be started on and each of the other hosts that the job will use.

```
rsh -l username-on-machine machine date
```

This command should print the date from the host *machine*. If you have a hosts file, you can automatically check all of the machines listed in it (see page 35).

Setting up Passwordless ssh Access to Remote Hosts

To use passwordless `ssh`, the machines involved must be specifically configured to allow it:

- An `sshd` server must be running on any machine to which you want to connect.
- RSA public key authentication needs to be enabled and empty passphrases must be allowed in the `sshd` configuration on any machine to which you want to connect.

Note: Public key authentication is enabled in OpenSSH by default.

To set up passwordless `ssh` access between hosts that share your login directory:

1. Generate your public/private RSA key pair:

```
cd ~/.ssh
ssh-keygen -t rsa
```

Note: When asked for a passphrase *do not* enter one; just press ENTER. If you specify a passphrase it defeats the purpose of configuring passwordless ssh.

2. Add your public key to the list of keys allowed to log in to your account:

   ```
   cat id_rsa.pub >> authorized_keys
   cat id_rsa.pub >> authorized_keys2
   ```

   The two separate files are necessary to support both OpenSSH 1.5 and OpenSSH 2.0 protocols. Some versions use just one or the other of these files.

3. Suppress the confirmation dialog you ordinarily get when you connect to a machine for the first time:

   ```
   echo "StrictHostKeyChecking no" >> ~/.ssh/config
   ```

   This is necessary if you want to use ssh non-interactively and you can not get RSA signatures for every host to which you want to allow connections in your known_hosts file ahead of time.

4. Clear your known_hosts file:

   ```
   rm known_hosts*
   ```

   This is necessary so that the new RSA key-pair mechanism is used for every host. Otherwise, hosts to which you previously connected using passwords might not use the new system automatically.

5. Make sure your home directory cannot be written by anyone but you:

   ```
   chmod go-w ~
   ```

   This is required before ssh will allow passwordless access to your account.

6. To ensure that Job Control uses ssh instead of rsh, you also need to set the environment variable SCHRODINGER_RSH to ssh:

csh, tcsh:	`setenv SCHRODINGER_RSH ssh`
bash, ksh:	`export SCHRODINGER_RSH=ssh`

   This only needs to be done on the machines where you want to use ssh instead of rsh. For instance, you might want to continue using rsh on your LAN and force ssh to be used only on your cluster. The easiest way to force ssh to be used on just certain hosts is to add the following line to the entries for those hosts in your schrodinger.hosts file:

   ```
   env:  SCHRODINGER_RSH=ssh
   ```

Preparing for Batch Queue Submission

Schrödinger products now provide basic support for submitting jobs to batch queues. Schrödinger supplies support for the PBS, LSF, and Sun Grid Engine (SGE) queueing systems in the standard software installation. Enabling batch queue submissions to a supported queueing system only requires the addition of a few lines to the `schrodinger.hosts` file and the specification of the queueing system and the queue name. These additions are described in the next subsection. For SGE, you may also need to set the `QPROFILE` variable in the `config` file to point to a file that sets up the environment for the queuing software—see page 40.

It should be reasonably straightforward to configure a Schrödinger software installation to support other queueing systems as well. The components required to support a batch system are a few text files that can be added or modified after installation. The nature of these files is explained in the following subsection.

If you intend to run distributed jobs on a cluster that is set up with batch queues, you should ensure that jobs can be submitted to the queues from a compute node on the cluster.

Configuring the schrodinger.hosts File for Batch Queues

To enable job submissions to a batch queue on a supported queueing system, you must add host entries to the `schrodinger.hosts` file that define the available queues. The command syntax is described in Table 2 on page 35. A sample of the host entries to be inserted into the `schrodinger.hosts` file is shown below:

```
# Batch submission to 'bigjobs' queue under PBS
Name: bigq
Host: cluster
Queue: PBS
Qargs: -q bigjobs
tmpdir: /storage/TMPDIR
processors:  8
#
# Batch submission to 'shortjobs' queue under PBS
Name: shortq
Host: cluster
Queue: PBS
Qargs: -q shortjobs
tmpdir: /storage/TMPDIR
```

This example defines two entries named bigq and shortq to which jobs can be sent on the host cluster. The job control facility distinguishes batch queues from hosts by the presence of the Queue: setting, which specifies the queuing system. This setting must be set to the name of the subdirectory of $SCHRODINGER/queues that contains the support files for the queuing system. The Qargs: setting specifies command line arguments for the queuing system's job submission command; for PBS, for instance, this is the qsub command.

You must also include a Host: setting because the Name: setting is used for the queue name. Like normal remote host entries, host entries for batch queues inherit settings made in the localhost entry of the schrodinger.hosts file. Batch queue entries can also have any of the other settings that host entries have, such as schrodinger: and tmpdir:. For queues on clusters, the tmpdir specifier is required and should refer to a directory that is available to all the nodes and writable by all users who will use that queue. On shared memory machines, the tmpdir specifier is optional.

Adding Support for an Unsupported Queueing System

To allow job submission to a batch queueing system, the Schrödinger Job Control facility requires these text files to be installed on the submission host:

1. A submit script, which is a wrapper for the queueing system's own job submission utility (qsub for PBS, bsub for LSF).
2. A cancel script, which is a wrapper for the queueing system's job removal command (qdel for PBS, bkill for LSF).
3. A config file, which contains settings for QPATH, QSUB, QDEL, QSTAT, and QPROFILE. This is the only file users should have to change. The default submit and cancel scripts are defined in terms of these settings. For example, $SCHRODINGER/queues/PBS/config contains the settings:

   ```
   QPATH=/usr/local/pbs/bin
   QSUB=qsub
   QDEL=qdel
   ```

   The QPROFILE variable specifies the absolute path on the queue host of a configuration file that needs to be sourced to set up the environment to use the queue. This variable is useful for setting up an environment for the queuing system that does not affect the global environment.
4. A template.sh file, which is a template for the shell script that is actually submitted to the batch queue and used to launch your calculation on the compute host.

These files are installed in a subdirectory of the `$SCHRODINGER/queues` directory. The name of this subdirectory is used as the name of the queueing system for the purposes of the `schrodinger.hosts` file, as described above. The standard software installation now creates PBS, LSF, and SGE directories in `$SCHRODINGER/queues`. These directories contain `submit`, `cancel`, `config`, and `template.sh` files for the PBS, LSF, and SGE queuing systems.

To modify these files or to provide new ones for an unsupported queueing system, it is necessary to understand what Job Control requires from each one. Each of the scripts is discussed below.

The `submit` Script

The `submit` script needs to support the command line syntax:

`submit` *job-script* `[`*qsub-options*`]`

where *job-script* is the name of a shell script that starts a job on the queue. This is always the first (and possibly only) command line argument to `submit`. Anything else on the command line must be passed on as arguments to the actual job-submission command.

If job submission is successful, `submit` should extract the batch ID from the output of the underlying job-submission command and report it in its output, in the form:

`BatchId:` *batchid*

If job submission fails for some reason, the script should exit with a non-zero exit code.

If you are creating your own `submit` script to support a new queueing system, you can use the `submit` scripts provided for PBS, LSF, and SGE as templates.

The `cancel` Script

The `cancel` script must support the command line syntax:

`cancel` *batchid*

where *batchid* is a batch ID assigned by the queueing system. The Schrödinger job control system keeps track of the batch ID of each submitted job so that the ID can be used for cancelling jobs. The `cancel` script should probably also return a nonzero exit status if the operation fails, but at present, the job control system ignores the exit status.

The Job Script Template File

The `template.sh` file is a skeleton for the Bourne-shell script that is actually submitted to the batch queue. The Schrödinger job-launching mechanism reads this file and inserts the commands necessary to launch the user's job, and then submits the resulting file to the queueing system using the `submit` command described above.

The following information from the `template.sh` file supplied for the LSF system illustrates how the `template.sh` file works.

```
#!/bin/sh
#BSUB -J %NAME%
#BSUB -o %DIR%/%JOBID%.qlog

export SCHRODINGER_BATCHID
SCHRODINGER_BATCHID=$LSB_JOBID

%ENVIRONMENT%

%COMMAND%
```

The `#BSUB` lines are directives that are interpreted by LSF. In this case, the first directive sets the LSF job name for this job to the Schrödinger job name, while the second specifies that any output from the job submission script should go to the file *jobid*`.qlog` in the directory from which the job was launched. Most other queueing systems also allow directives to be provided in the initial comment lines of the job submission scripts.

The words delimited by percent signs are variables, which are replaced at job launch time with the actual job name, Schrödinger job ID, etc., for the job you are submitting. Variables that you can put in any new `template.sh` file are listed in the following table:

Table 3. Batch script variables.

Variable	Variable action
`%NAME%`	Schrödinger job name, usually derived from your input file name.
`%DIR%`	Directory from which the job was submitted.
`%HOST%`	Machine from which the job was submitted.
`%USER%`	Name of the user who submitted the job.
`%JOBID%`	Job ID assigned by the Schrödinger job control system.
`%ENVIRONMENT%`	Commands which define environment variables that are required for your job to run.
`%PRODUCT%`	Product name (NOT the executable).
`%APP_EXEC%`	The name of the exec variable for the product.
`%VER_ARGS%`	Version arguments.

Table 3. Batch script variables.

Variable	Variable action
`%HUNT_PATH%`	The path to the hunt executable.
`%JOBDB%`	The path to the job database.
`%NPROC%`	Number of processors that were requested.
`%LOGDIR%`	The directory in which log files are written.
`%HOME%`	Home directory on the submission host.
`%COMMAND%`	Command that launches the Schrödinger `jmonitor` program, which sets up, runs, and cleans up after your calculation.

The `%ENVIRONMENT%` and `%COMMAND%` lines are the only lines that are absolutely required in this script and they must appear in this order. These variables are assigned by the job control system and are not configurable by the user.

The final component of this script is the two-line section that sets the `SCHRODINGER_BATCHID` environment variable to the actual batch ID assigned to this job. The batch ID is usually provided by the queueing system in a special environment variable such as the `LSB_JOBID` variable used by LSF. The `jmonitor` program checks for the `SCHRODINGER_BATCHID` environment variable and saves the batch ID in the job record, where the user can look it up.

If you want to run parallel Jaguar jobs, you should ensure that the path to `mpirun` or `poe` is prepended to the `PATH` environment variable, by adding the following line after the setting of the batch ID:

`PATH=`*relevant-path*`:$PATH`

Additional Information

For additional information about PBS, LSF, and SGE, see the following web sites:

PBS: http://www.openpbs.org

http://www-unix.mcs.anl.gov/openpbs

LSF: http://www.platform.com/products/wm/LSF

SGE: http://gridengine.sunsource.net

Please see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

Configuring Clusters

The configuration of a cluster to run Schrödinger software must take into account the special issues of communication between the compute nodes, the manager nodes, and the external network[1], and the impact that this communication might have on performance.

Like any other host, each compute node must have access to a license, the software and the job-related files. Special issues for clusters are discussed for each of these below.

- **Access to a license.** The compute nodes must be able to read the license file and be able to open a socket on the license server. The license file can be stored on the external network, the internal network, or copied to each node. Since this file is small, the location does not matter.
- **Access to the installed Schrödinger software.** To reduce network traffic, Schrödinger software should be installed either on each compute node's local disk, or on a file system that is accessible internally to all cluster nodes (that is, one that does not create network traffic through the manager node to the external network).
- **Access to files related to the job.** If the files cannot be read directly by the compute node, the Job Control facility uses passwordless `ssh` or `rsh` to transfer data to and from the job submission host, or to an intermediate host that has access to the relevant file systems of the job submission host. Without this capability, input and output may have to be copied by hand, jobs cannot be monitored, and incorporation of results into Maestro projects cannot be done automatically. Access to files can be enabled in the following ways:
    - Mount the relevant file systems of the job submission host on the compute nodes.
    - Enable passwordless `ssh` between the job submission host and the compute nodes.
    - Enable passwordless `ssh` between the manager nodes and the compute nodes, and mount the relevant file systems of the job submission host on the manager nodes.

  You can also use passwordless `rsh`, but `rsh` has access to a limited number of ports and is more likely to result in job failure as a consequence. See "Setting up Passwordless ssh Access to Remote Hosts" on page 37 for more information.

[1] that is, the connection to hosts outside the cluster

If passwordless ssh is not permitted between the job submission host and the compute nodes, the firewall that prevents the compute nodes from connecting to external hosts via ssh must be configured to reject, rather than drop, such traffic.

The preferred option is to allow job submission to the cluster from the network, rather than restricting it to submission from the manager node. A summary of the requirements is given in Section 3.4 of the *Job Control Guide*.

To run parallel Jaguar, the following are required in addition:

- The user's home directory must be mounted on the compute nodes.
- The user's home directory must contain a .rhosts file listing all the compute nodes.
- Passwordless ssh has to be enabled between compute nodes.

To optimize the performance of a cluster for Schrödinger software, we suggest that you consider the following options when purchasing, upgrading, or configuring a cluster:

- Invest in a highly capable file server for the external network.
- Invest in shared storage for the private (intra-cluster) network, to reduce traffic to and from the external network. Shared storage makes installation and maintenance of the code much simpler, and can be used to store large data files, either temporarily or on a long-term basis.
- Divide services among several management nodes. For example, the queueing system, the private network's shared storage and the routing could all be handled by separate management nodes.
- Ensure that the management nodes have fast processors, large memory, and high-quality motherboards and network interfaces.
- Run more recent Linux versions such as Red Hat 9.0 or SUSE 9.0, which have better facilities for NAT and related functionality than earlier versions.
- Run a robust queuing system that is relatively immune to stalling, crashing or bringing down its host if it is heavily loaded.

Preparing for Grid Computing

To run Schrödinger products under a grid computing system, you need to install the product on both the Linux workstations from which you want to submit jobs *and* on the grid server. The Schrödinger grid product installation package contains the files for both. The general installation workflow is as follows:

1. Install the Linux version of the Schrödinger product. This provides the script that installs the infrastructure module on the grid.
2. Install the infrastructure module on the grid. Schrödinger supports the following grid infrastructures:
    - LSF Desktop from Platform Computing (see "Installing for LSF Desktop" below)
    - GridMP from United Devices (see "Installing for GridMP" on page 47)

The instructions that follow are general instructions for installing grid infrastructure modules.

Installing for LSF Desktop

To run Schrödinger products under Platform Computing's LSF Desktop grid server (formerly called ActiveCluster, AC), you must *first* install the product on the Linux workstations from which you want to submit jobs (see "Installation Process Summary" on page 10) *then* install the ActiveCluster program module (see below).

1. Create a new directory to store the AC program module:

    ```
    mkdir $SCHRODINGER/product-vversion/bin/WIN32-x86
    ```

2. Move the AC program module to the directory you just created:

    ```
    cp product-vversion-WIN32-x86.zip $SCHRODINGER/product-vversion/bin/
    WIN32-x86
    ```

3. Create an entry for the LSF Desktop server in your `schrodinger.hosts` file by adding the following lines:

    ```
    # ActiveCluster server
    name:     ac
    host:     localhost
    platform: WIN32-x86
    queue:    AC
    qargs:    -W 6:00
    ```

As for other entries, the `name` setting is arbitrary: it defines the “virtual hostname” that users specify to direct their jobs to the grid.

LSF Desktop is simply a special type of LSF queue, so its configuration is just like any other queue configuration. You must specify AC for the `queue` setting, but you can use the `qargs` setting to specify arguments to be passed to the LSF Desktop queuing system. You can include multiple entries with different `qargs` settings.

Installing for GridMP

To run Schrödinger products under GridMP, you must *first* install the product on the Linux workstations from which you want to submit jobs (see “Installation Process Summary” on page 10) *then* install the GridMP program module (see below).

If you want to run jobs using GridMP on Linux workstations, you must also install the software on these workstations, preferably on a file system that is accessible to all of them.

1. Create an entry for the GridMP server in your `schrodinger.hosts` file by adding the following lines:

   ```
   # GridMP server
   name:     gridmp
   mpconfig: uduserconf
   ```

   The `name` setting is arbitrary: It defines the “virtual hostname” that users specify to direct their jobs to the grid.

   The `mpconfig` setting specifies the name of a configuration file containing the contact information for the grid server. The launch scripts look in the current directory and the user’s home directory for a file with this name when they need to contact the server. If you specify it using an absolute path name, only that location is checked.

   You can include the contact information for the grid server directly in the hosts file, by adding the following settings:

   `mpserver:` *grid-server-URL*
   `mpfiler:` *file-server-URL*
   `mpuser:` *account-name*
   `mppassword:` *account-password*

   If you also add an `mpconfig` setting, the values in the configuration file override the values in the hosts file.

2. Create a configuration (`uduserconf`) file specifying the URLs for the `mgsi` and file servers, and the username and password needed to log on to the server.

This file should have the same format as the configuration file for the sample scripts in the UD SDK. (There is a sample `uduserconf` in the installation directory.) The four parameters for access to the grid server are specified using the lines:

```
MGSI_SOAP_URL = grid-server-URL
MGSI_FILESVR_URL = file-server-URL
MGSI_USERNAME = account-name
MGSI_PASSWORD = account-password
```

If the `mpconfig` setting in the `schrodinger.hosts` file gives an absolute path, the `uduserconf` file should be copied to the location specified. If the `mpconfig` setting does not include a path, users must copy the `uduserconf` file to their home directory (or working directory) to be able to use the grid.

3. Install the *product*-v*version-platform*`.mod` program module on the grid by entering one of the following commands:

```
$SCHRODINGER/utilities/mpinstall product-vversion-platform.mod
```

or

```
$SCHRODINGER/utilities/mpinstall -c configfile
    product-vversion-platform.mod
```

Here, *platform* is `WIN32-x86` for Windows hosts or `Linux-x86` for Linux hosts. You should run this command from the directory containing the *product*-v*version-platform*`.mod` file. It creates an application and program named *product* and installs the given file with the version number *version*.

Note: If you do not specify the configuration file explicitly using the `-c` option, `mpinstall` looks for a `uduserconf` file to get the URL and account information for the grid server.

Launching Maestro

To launch Maestro, first check that your `DISPLAY` and `SCHRODINGER` environment variables have been set (page 21), then enter the command:

```
$SCHRODINGER/maestro
```

If the `maestro` startup script was aliased or `$SCHRODINGER` was added to your `PATH` environment variable, you can omit `$SCHRODINGER/` from the command. Options for the `maestro` command are given in Section 2.1 of the *Maestro User Manual*.

If you have difficulty launching Maestro, send the input and output from the failed `maestro` command in an e-mail message to help@schrodinger.com. Include in your message the type of workstation, the operating system version, and the Maestro version you are using. The required information is displayed by the command

```
$SCHRODINGER/machid
```

The various computational programs can be launched from Maestro by selecting the program from the Applications menu.

Hardware and Software Requirements

This section lists the general minimum system requirements for Schrödinger products. If the product-specific requirements differ from those listed below, they are given in the "Product Notes" on page 55. For each product, the platforms that are supported have a corresponding entry in Table 4 on page 53.

All Platforms

- Perl version no earlier than 5.004
- `gunzip`
- 256 MB memory minimum, 1 GB recommended.
- 4 GB scratch disk space minimum; 60 GB recommended, at 10000 RPM.
- For stereo viewing, a monitor with a refresh rate of 100 Hz or more is recommended.

Linux

Supported operating systems:

- RHEL 3.0, 4.0, and 4.1, the 4.x versions installed with back-compatibility glibc options
- SUSE 9.0
- CentOS 4.0, and 4.1 installed with back-compatibility `glibc` options
- Red Hat Linux 7.3 and 9.0

Incompatible operating systems:

- SUSE 9.1
- Red Hat Linux 8.0

Other possibly compatible operating systems:

Schrödinger products in Schrödinger Suite 2006 will probably run correctly on the following platforms, but they are not supported:

- RH 7.1, provided users have updated `glibc` to version 2.2.4-33
- RHEL 2.1 installed with back-compatibility `glibc` options
- SUSE 8.0, SUSE SLES 9 installed with back-compatibility `glibc` options
- Fedora Core 3, 4 installed with back-compatibility `glibc` options
- Fedora 1.0

- Any installation of Linux with a 2.4 kernel or a 2.6.9 kernel having the following minimum set of software packages installed:
    - XFree86 4.2.0
    - freetype 2.0.9
    - glibc 2.2.5
    - lesstif 0.93.18
    - libjpeg 6b
    - libstdc++ 2.96
    - libtiff 3.5.7
    - zlib 1.1.3

Hardware:

- x86-compatible processor, such as a Pentium family processor (including Pentium-4 and Xeon), AMD K6, Athlon or Opteron. Our executables are supported on the Opteron under either 32- or 64-bit operating systems. Larger cache sizes result in improved performance of most Schrödinger software.
- Linux-supported network card with a configured network interface

SGI IRIX

Operating system:

Minimum requirements are as follows.

- IRIX 6.5.19.
- Run-time environment version 7.4. Earlier versions have library incompatibilities. For parallel Jaguar execution, you must install version 7.4.2m.
- The `libscs` library (Scientific Computing Software Library) must be installed. You can download this library from http://www.sgi.com/products/evaluation/6.5_scsl_1.4.1.3/[1]. The `libscs` library includes what was formerly in `libblas`. Some products require `libblas` in addition to `libscs`.

  To check for `libscs`, enter the command:

  ```
  showprods | grep -i scsl | grep -v 64bit
  ```

Hardware:

- MIPS 4 or compatible processor

[1] Please see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

SGI Altix

Jaguar is the only product currently supported on this platform.

Supported operating systems:

- SUSE 9.0
- RHEL 3.0; RHEL 4.0 should run Jaguar but is not supported.

Hardware:

- Itanium 2 processor

IBM AIX

Operating system:

- AIX 5.2. The command `oslevel -r` should report at least `5200-02`.
- Fortran run-time library version 8.1.1.0. The command `lslpp -l xlfrte` should report at least `8.1.1.0`.

Hardware:

- At least Power3 or compatible processor.

Schrödinger products will probably run correctly on the pwr5 processor and whatever OS revision it runs under, but it is not supported.

Disk Space

Approximate disk space requirements in MB for the installation of each product are given in Table 4 on page 53. Disk space requirements for the `data` modules are included in the values given in the table. The `data` modules only need to be installed once for each installation. Disk space for mmshare, Python, and services are not included in the values for the products, as they only need to be installed once for all products. The mmshare and Python modules are automatically installed. Disk space for the documentation amounts to only a few megabytes per product; the entire documentation set occupies about 60 MB.

Table 4. Disk space requirements in MB for installation of Schrödinger software.

Product	Linux	SGI IRIX	SGI Altix	IBM AIX	Common
Maestro, Strike[a]	139	136			69
mmshare[b]	131	134	160	105	18
Python[b]	100	95		75	9
services	38	28		11	
CombiGlide	69	60			26
Epik	42	33		15	
Glide, Liaison, QSite[c], SiteMap	126	114		117	77
Jaguar[d]	127	207	404	114	
MacroModel, LigPrep[e]	205	198		169	139
Phase	134	199			7
Prime	1320	1328			1288
BLAST[f] HMMER/PFAM[f]	3452	3462			
PDB[g]	4506	4506			
Prime-CM	439	444			416
QikProp	40	29		9	1

a. Strike is contained in Maestro.
b. Automatically installed with every product
c. Does not include Jaguar disk space.
d. Includes parallel binaries
e. LigPrep is contained in the MacroModel installation
f. Third party software
g. Third party database only - no software

Maestro 7.5 Requirements

Supported Platforms

Maestro is supported on SGI IRIX and Linux x86 platforms.

System Requirements

All Platforms

Minimum requirements are as follows:

- An X11R6 X server on any machine to which Maestro is displayed.
- X Servers must include the GLX OpenGL extension, and OpenGL must be enabled.
- For stereo viewing, a monitor with a refresh rate of 100 Hz or more is recommended. Most LCD displays do not have a sufficiently high refresh rate.

SGI

- X/Motif
- Installed n32 libraries: Xm (Motif), Xt, Xext (X extensions), X11, GLU, GL, ftn, ftn90, fortran, libscs

Linux

Minimum requirements are as follows:

- Graphics: 16-bit color while running in 1280 x 1024 resolution. Lower resolutions are supported, but the minimum is 16-bit color. A graphics card that supports hardware-accelerated OpenGL is strongly recommended.
- Stereo viewing: a graphics card that supports quad-buffered stereo. To run Maestro in hardware stereo mode, you must edit `/etc/X11/XF86Config-4` or `/etc/X11/XF86Config` to set the driver in stereo-capable mode.

Product Notes

This section contains notes specific to each product. These notes give information in addition to the requirements listed above.

CombiGlide 1.0

CombiGlide makes use of both Glide and LigPrep, which must be installed along with CombiGlide. LigPrep licenses must be obtained in addition to CombiGlide licenses.

Jaguar 6.5

1 GB scratch disk space minimum per process is recommended. Large jobs, such as frequency and LMP2 calculations, can use several gigabytes of scratch disk space.

Use local disks for scratch space. Performance is significantly reduced if an NFS-mounted scratch disk is used. Do not use scratch directories that are symbolic links, because this is known to prevent Jaguar jobs from running, especially under Linux.

Jaguar can be run in parallel under MPI. Information on system setup for MPI parallel execution is given in "Jaguar 6.5 Parallel Execution" on page 57.

LigPrep 2.0

LigPrep requires the installation of both `libblas` and `libscs` on SGI IRIX platforms.

MacroModel 9.1

MacroModel requires the installation of both `libblas` and `libscs` on SGI IRIX platforms.

The XCluster GUI is shipped with the IBM AIX distribution and should work, but is not supported. There are no special platform-specific requirements.

Phase 2.0

Phase makes use of LigPrep and MacroModel features, and the Phase license covers the use of these features. Phase requires the installation of the MacroModel, mmshare, and services modules, in addition to the Phase module.

Prime 1.5

Prime requirements include requirements for Prime-CM and Induced Fit Docking. Minimum memory required is 512 MB, 1 GB is recommended.

The data modules for Prime span two CDs. If you are installing from CD, you should install from CD 1 first, then CD 2.

Prime requires the installation of or access to various third-party products. See "Prime Third-Party Software and Databases" on page 63 for more information.

Prime requires the installation of both `libblas` and `libscs` on SGI IRIX platforms.

Python 3.0

The Python 3.0 package includes a set of Python scripts along with the standard Python v2.3.3 distribution. You must also install Maestro if you plan to run Python modules with Maestro-related functionality. This package is installed by default with all products.

QSite 4.0

QSite uses Jaguar for the QM part of the calculations. See "Jaguar 6.5 Parallel Execution" on page 57 for information on parallel Jaguar requirements.

SiteMap 2.0

Installation of `Xvfb` can be beneficial, though not necessary, for SiteMap operation. The latest `Xvfb` executables for some UNIX and Linux operating systems can be obtained from[1]:

http://ftp.xfree86.org/pub/XFree86/4.5.0/binaries

The executable should be installed and added to the default `PATH`. More detail is available in Appendix A of the *SiteMap User Manual*.

[1] Please see the notice regarding third party programs and third party Web sites on the copyright page at the front of this manual.

Jaguar 6.5 Parallel Execution

Parallel Jaguar is available for all platforms and included in the executable set. To run in parallel you must have the Message Passing Interface (MPI), which is contained in MPT for SGI, MPICH for Linux, and POE for IBM. Jaguar runs on shared-memory architectures in SMP mode, or on distributed architectures and clusters, such as IBM SP2 and Beowulf.

Parallel Execution Requirements

SGI (IRIX and Altix)

- Message-Passing Toolkit (MPT) no earlier than 1.6.0.0 for IRIX and 1.10 for Altix.
- Array Services no earlier than 3.5 for IRIX and 3.7 for Altix.
- Runtime library no earlier than 7.4.2m.

Linux

- MPICH1
- The kernel must be compiled for SMP (symmetric multiprocessing).

IBM AIX

- POE version no earlier than 4.1.0.0
- If LoadLeveler is used, a version no earlier than 2.1

Installing Tools for Parallel Execution

After installing Jaguar, edit the `schrodinger.hosts` file in the `$SCHRODINGER` directory, and make an entry for each host on which parallel Jaguar jobs will be run. See "Modifying the Hosts File" on page 34 for details.

Each user must create a `.rhosts` file in his or her home directory that contains the name of each host on which parallel Jaguar jobs will be run, followed by the user's login name. See "Setting Up Passwordless rsh Access to Remote Hosts" on page 36 for more information.

To run parallel Jaguar jobs from a batch queue, you should ensure that the path to `mpirun` or `poe` is prepended to the `PATH` environment variable in the template script for the batch queue. See page 43 for more information.

SGI Installation

You must install the Message-Passing Toolkit (MPT) and Array Services. The installation of these packages requires root permission. For Altix, both of these packages are installed from RPMs that are part of the SGI ProPack software package. For IRIX, you can download these packages from http://www.sgi.com/products/evaluation/[1]. The installation instructions are as follows:

1. Install the MPT package if it is not already installed. You can check to see if MPT is already installed with the following command:

   IRIX: `showprods | grep MPI`

   Altix: `rpm -q sgi-mpt`

2. Install Array Services if it is not already installed. You can check to see if Array Services is installed with the following command:

   IRIX: `showprods | grep arraysvcs`

   Altix: `rpm -q sgi-arraysvcs`

3. Start the array services daemon with the following command:

   ```
   /etc/init.d/array start
   ```

   The `arrayd` daemon can be configured to start automatically at system startup with the command

   ```
   chkconfig array on
   ```

LINUX Installation

For Linux, parallel Jaguar requires the MPICH package. If Jaguar is to run in parallel on a multiprocessor machine, the kernel must be compiled for SMP (symmetric multiprocessing).

Installing MPICH

We recommend building MPICH from the MPICH1 source code (*not* the MPICH2 source code). The latest source code is available from http://www-unix.mcs.anl.gov/mpi/mpich1[1]. Instructions for building and installing MPICH are included with the source code. When you build MPICH from the source code, include the following configure option:

```
--with-device=ch_p4
```

[1] Please see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

The directory in which you installed MPICH is referred to below as *MPICH-install*.

Configuration

1. Add the MPICH `bin` directory to the `PATH` environment variable. This is necessary for Jaguar to find the `mpirun` launch script.

   csh/tcsh: `setenv PATH` *MPICH-install*`/bin:$PATH`

   ksh/bash: `export PATH=`*MPICH-install*`/bin:$PATH`

2. Edit the file *MPICH-install*`/share/machines.LINUX` and list the names of the hosts available for parallel calculations. Each host name should be listed once for each processor in that host. The host name should match the output of the `hostname` command. For example:

   ```
   homer.mynet.edu
   homer.mynet.edu
   marge.mynet.edu
   marge.mynet.edu
   bart.mynet.edu
   ```

   Do not use the *hostname*:*ncpus* syntax, because this tells MPICH to use shared memory, and Jaguar does not use shared memory for MPI calculations under Linux.

3. Edit the `schrodinger.hosts` file in the directory where Jaguar was installed, and list in it the names of the hosts in the `machines.LINUX` file. The host names in `schrodinger.hosts` need not include the domain name. See "Modifying the Hosts File" on page 34 for details on the format of the `schrodinger.hosts` file. For the above example, the `schrodinger.hosts` file would look like:

   ```
   host:        homer
   schrodinger: /apps/Schrodinger
   tmpdir:      /scr
   processors:  2
   !
   host:        marge
   schrodinger: /apps/Schrodinger
   tmpdir:      /scr
   processors:  2
   !
   host:        bart
   schrodinger: /apps/Schrodinger
   tmpdir:      /scr
   processors:  1
   ```

4. Ensure that `rsh` is enabled. By default, Jaguar uses `rsh` to communicate with remote nodes (even if you are running on a stand-alone SMP workstation with 2 CPUs). To enable `rsh`, each user must create a file called `.rhosts` in his or her home directory. The `.rhosts` file should contain the name of each host listed in the file `machines.LINUX`, followed by the user's login name, e.g.,

   ```
   homer.mynet.edu username
   marge.mynet.edu username
   bart.mynet.edu username
   ```

   The `.rhosts` file must be owned by the user (not by root) and must not be writable by anyone except the user, or authentication fails. To ensure this, enter the command

   ```
   chmod 644 ~/.rhosts
   ```

   We strongly recommend that you test `rsh` connections by using the shell script `tstmachines`, which is in *MPICH-install*`/sbin`. This script attempts to run several `rsh` commands on each of the hosts listed in the file `machines.LINUX`, and lists any problems. If the command is successful it returns with no output. You can also use the `-v` option on the command line to see exactly what the script is doing.

   Note: Because MPICH uses rhosts authentication, you must set up the `.rhosts` file even if you are using `ssh` for communication.

Launching the Secure Servers

Jaguar relies on the MPICH secure server, `serv_p4`, to transport the environment to all nodes used in a parallel calculation. The secure server must be running on all computers on which Jaguar is to run in parallel, which is normally all hosts listed in the `machines.LINUX` file. The secure server uses a communication port that is specified by the user (or by root).

To launch the MPICH secure server, enter the command

```
$SCHRODINGER/utilities/mpich start -p port
```

The port number (*port*) should be a four-digit number greater than 1023. If `-p` *port* is not specified, the value of `MPI_P4SSPORT` is used for the port number. If `MPI_P4SSPORT` is not set, the default value of 1234 is used. Although each user may launch the secure server and select a port number for private use, we recommend that the system administrator launch the server as root so that all users can use the same port number. The port number should be different from the default 1234, to avoid conflicts with other uses of the secure server ports. The `mpich start` command launches the secure servers on all of the hosts listed in the `machines.LINUX` file.

To use the secure servers, the following environment variables must be set:

csh/tcsh:

```
setenv MPI_USEP4SSPORT yes
setenv MPI_P4SSPORT port
```

ksh/bash:

```
export MPI_USEP4SSPORT=yes
export MPI_P4SSPORT=port
```

The port number assigned to `MPI_P4SSPORT` must match the port number used to launch the secure server. These environment variables can be set up by root in the default environment, or they can be set up in a login script to avoid having to set them manually at each session. The last strategy does not work for `ksh`, which does not execute a login script.

An alternative script for managing the secure servers is described in Chapter 13 of the *Jaguar User Manual*. To launch the secure servers, enter the command

```
$SCHRODINGER/utilities/mpich start
```

Note: Since the secure servers are `setuid` processes, there is some security risk in launching them as root. MPICH is third-party software: see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

IBM Installation

For IBM, you need to install the Parallel Operating Environment (POE) package, which includes the MPI libraries. Jaguar requires a version of POE no earlier than 4.1. Be sure to check the README file in `/usr/lpp/ppe.poe` and the man page for details on POE. If you use LoadLeveler, it must be a version that is no earlier than 2.1.

You may need to set an environment variable in order to use multiple processors for a job. The variable to set depends on how your machine has been configured; specifically whether you are running the Job Manager or not. The Job Manager manages pools of nodes, and assigns specific parallel jobs to specific nodes. To test whether you are using the Job Manager, type

```
ps aux | grep jmd
```

If you see `jmd` processes listed, you are running the Job Manager. In this case, you need to tell Job Manager the pool from which you want to have nodes assigned to you. The command `jm_status -P` lists the available pools and their member nodes. The environment variable that sets your job pool is called `MP_RMPOOL`, and it should be set to the appropriate pool number:

csh/tcsh:

```
setenv MP_RMPOOL 1
```

ksh/bash:

```
export MP_RMPOOL=1
```

If your machine does not use the Job Manager, you can set the environment variable MP_HOSTFILE to the file that contains the host list. If MP_HOSTFILE is not set, then the host file is assumed to be called host.list and to reside in the current directory (see the poe man page). The host file should contain the names of the nodes on which parallel jobs can be run. The node name is listed once for each processor in that node.

Prime Third-Party Software and Databases

Required Third-Party Software and Databases

To use Prime and Prime-CM, you must install or have access to the PDB, the BLAST program and associated sequence databases, and the HMMER and Pfam programs and associated databases. The BLAST database format should conform to that generated with `formatdb` version 2.2.6. If you do not intend to identify families for your query sequence, you do not need to install the HMMER and Pfam programs. Disk space requirements are listed in Table 4 on page 53.

Note: For Induced Fit Docking, you do not need to install the third-party programs or databases.

The required third-party programs (BLAST, HMMER and Pfam) are provided on CD and in the Prime-CM download. The databases are provided on CD. If you downloaded Prime-CM and do not have the databases, instructions for obtaining them are provided in the Support Center of our web site.

If you install these third party products from the CDs supplied by Schrödinger, you must run the `INSTALL` script for each CD. Do not change CDs while the `INSTALL` script is being executed: the script will fail. If you install these third-party products in a location other than the default location, `$SCHRODINGER/thirdparty`, you must set the environment variable `SCHRODINGER_THIRDPARTY` to the chosen location.

If you already have copies of the third-party products, you can provide links to them using the environment variables described in the table below. You do not need to set the environment variables listed below if you are installing Prime from the Schrödinger-supplied CDs.

You can set the environment variables for remote hosts in the `schrodinger.hosts` file (see "Modifying the Hosts File" on page 34 for more information).

Table 5. Environment variables defining the location of third-party software and databases (both required and optional) for Prime.

Environment Variable	Description
`SCHRODINGER_PDB`	PDB distribution directory (contains the `data` directory).
`PSP_BLASTDB`	BLAST database directory (contains databases `nr` and `pdb`)
`PSP_BLAST_DIR`	BLAST executable directory
`PSP_BLAST_DATA`	BLAST matrices directory
`PSP_HMMER_DIR`	HMMER executable directory
`PSP_HMMERDB`	Pfam database directory
`PSP_PSIPRED_DIR`	PSIPRED installation (contains `bin` and `data` directories)
`PSP_PSIPRED_DB`	PSIPRED sequence database
`PSP_SSPRO_DB`	SSPRO sequence database

Information on the third-party software and databases can be found at the following locations:

BLAST: http://www.ncbi.nlm.nih.gov/blast

HMMER: http://hmmer.wustl.edu

Pfam: http://www.sanger.ac.uk/Software/Pfam

PDB: http://www.rcsb.org/pdb

Please see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

Note: The Prime third party installation does not include obsolete PDB structures. In general we recommend updating the PDB directory after installing third party software, and have provided an update script for this purpose:

```
$SCHRODINGER/utilities/rsync_pdb
```

We have also provided a script for updating and formatting the BLAST database:

```
$SCHRODINGER/utilities/update_BLASTDB
```

Optional Third-Party Software and Databases

In addition to the secondary structure prediction program bundled with Prime, there are third-party secondary structure prediction programs that can be downloaded from their respective web sites and used with Prime. For the latest information on these programs, visit the Support Center of our web site. The Support Center also includes installation instructions for these programs.

Please see the notice regarding third party programs and third party web sites on the copyright page at the front of this document.

Copyright Notices

NCSA HDF5 Software Library and Utilities

Copyright Notice and Statement for NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 1998, 1999, 2000, 2001, 2002, 2003, 2004 by the Board of Trustees of the University of Illinois
All rights reserved.

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC), Lawrence Livermore National Laboratory (LLNL), Sandia National Laboratories (SNL), Los Alamos National Laboratory (LANL), Jean-loup Gailly and Mark Adler (gzip library).

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.
3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.
4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and to credit the contributors.
5. Neither the name of the University nor the names of the Contributors may be used to endorse or promote products derived from this software without specific prior written permission from the University or the Contributors, as appropriate for the name(s) to be used.

6. THIS SOFTWARE IS PROVIDED BY THE UNIVERSITY AND THE CONTRIBUTORS “AS IS” WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall the University or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

DISCLAIMER: This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

C and C++ Libraries for Parsing PDB Records

Technical Support

Many of the common issues with installation, licensing, and running of Schrödinger software are addressed on the Frequently Asked Questions pages, at https://www.schrodinger.com/General_FAQ.html.

If you experience difficulty installing your Schrödinger software, or if you have any other questions, contact Schrödinger using the information below. Generally, e-mail correspondence is best because you can send relevant machine output.

Schrödinger
E-mail: help@schrodinger.com
USPS: 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532
WWW: http://www.schrodinger.com

When sending e-mail messages, please include the following:

- all relevant user input and machine output
- purchaser (company, research institution, or individual)
- computer platform type
- operating system with version number
- software version number
- Maestro version number
- mmshare version number

Much of the machine and system information listed above can be produced by entering the following command:

```
$SCHRODINGER/machid
```

If you have trouble with licenses, please also send the output from the commands listed in "Requesting assistance" on page 30.

Index

120 West 45th Street
32nd Floor
New York, NY 10036

101 SW Main Street
Suite 1300
Portland, OR 97204

3655 Nobel Drive
Suite 430
San Diego, CA 92122

Dynamostraβe 13
68165 Mannheim
Germany

QuatroHouse, Frimley Road
Camberley GU16 7ER
United Kingdom

SCHRÖDINGER®